

Tomasz GRATKOWSKI

UNIwersytet Zielonogórski, Instytut Informatyki i Elektroniki

Zastosowanie klauzul w formalizacji czynności w metodyce Cheesmana-Danielsa

Mgr inż. Tomasz GRATKOWSKI

Mgr inż. Tomasz Gratkowski skończył studia na Wydziale Elektrycznym Politechniki Zielonogórskiej, kierunku Elektrotechnika, specjalność Inżynieria Systemów Komputerowych. Jest pracownikiem Instytutu Informatyki i Elektroniki Uniwersytetu Zielonogórskiego oraz współpracuje z firmami z branży informatycznej. Głównym tematem zainteresowań naukowych są projektowanie systemów informatycznych, metodyki używane w inżynierii oprogramowania oraz systemy rozproszone.

e-mail: T.Gratkowski@iie.uz.zgora.pl



Streszczenie

Artykuł przedstawia formalizację czynności tworzenia modelu typów na etapie specyfikowania z zastosowaniem języka klauzul. Zastosowanie języka logiki jest bliższe ludzkiemu myśleniu niż inne formalizmy. Umożliwia to przeniesienie mechanizmu wnioskowania dokonywanego przez analityka systemowego w procesie projektowania systemu, do zapisu czytelnego dla systemów komputerowych. Opisana reprezentacja zostanie użyta podczas opracowywania systemu wspomagającego analityka w podejmowaniu decyzji na poszczególnych etapach projektu.

Słowa kluczowe: Rational Unified Process, metodyka Cheesmana-Danielsa, formalizacja, klauzule, komponenty, przetwarzanie symboliczne.

Using clausal in formalization activities from Cheesman-Daniels methodology

Abstract

This paper presents a formal representation of develop type model activity at specification stage in a clausal language. Logic language is more human-oriented than other formalism. Using clauses enables to represent a mechanism of concluding used in a process of designing the system. This representation will be used in a decision support system which will support a designer.

Keywords: Rational Unified Process, Cheesman-Daniels methodology, formalisation, clausal, components, symbol manipulation.

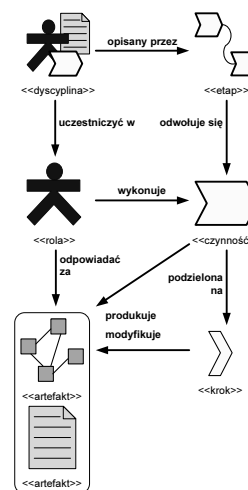
1. Wstęp

Duże projekty informatyczne są często realizowane w oparciu o technologie komponentowe. W ramach takiego projektu realizowana jest duża liczba różnych czynności, które wykonują członkowie zespołu w celu zrealizowania zdefiniowanego w umowie z klientem systemu informatycznego. Dlatego na przełomie lat opracowano wiele metod wspomagających proces projektowania, budowania, testowania oraz konserwowania oprogramowania [3].

W książce [1] Cheesman i Daniels opisali metodykę bazującą na metodach Rational Unified Process (RUP) [5] oraz Catalysis [2], w której autorzy skoncentrowali się na specyfikacji komponentów z wykorzystaniem modeli UML [7]. Metodyka Cheesmana-Danielsa (dalej metodyka CD) została podzielona na sześć dyscyplin. Każda dyscyplina składa się z etapów, w skład których wchodzi czynności, które podzielone są na kroki. Na rysunku 1 przedstawiono zależności pomiędzy głównymi elementami wchodzącymi w skład metodyki CD. W procesie przepływu czynności analityk systemowy wykonujący poszczególne zdefiniowane w czynnościach kroki operując na artefaktach [5]. Artefakty są efektem wykonania poszczególnych kroków oraz źródłem niezbędnych informacji dla wykonania kroków. Cały proces projektowania systemu jest realizowany przez człowieka. Analityk systemowy wykorzystując swoją wiedzę, doświadczenie oraz

informacje zawarte w artefaktach wejściowych, wykonuje poszczególne kroki, umożliwiające zaprojektowanie systemu informatycznego.

Prowadzone badania nad automatycznym dowodzeniem twierdzeń przy użyciu komputerów, przyczyniły się do opracowania języka klauzul i związanych z tym językiem logiki systemów wnioskowania. Zastosowanie języka logiki, jako modelu komunikacji z komputerem, jest bliższe ludzkiemu myśleniu niż inne formalizmy [4].



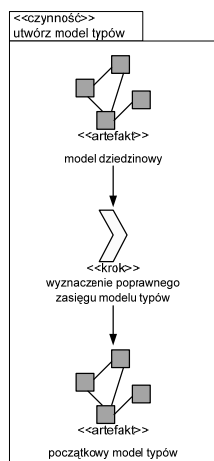
Rys. 1. Zależności pomiędzy elementami użytymi w metodyce CD w notacji Software Process Engineering Metamodel (SPEM) [6]

Fig. 1. Process elements used in CD methodology in Software Process Engineering Metamodel (SPEM) notation [6]

Wykorzystanie języka klauzul do formalizacji sposobu wnioskowania wykonywanego podczas wykonywania poszczególnych kroków w metodyce CD jest wstępnym etapem prowadzonych badań. Prowadzone przez autora badania mają na celu opracowanie mechanizmu automatyzacji wybranych kroków realizowanych w metodyce CD. Prezentowany artykuł przedstawia formalizację pierwszej czynności z etapu specyfikowania przy użyciu klauzul.

2. Formalizacja czynności tworzenia modelu typów w metodyce Cheesmana-Danielsa przy użyciu klauzul

W metodyce CD główny nacisk położono na dyscyplinę specyfikowania. Dyscyplina specyfikowania została podzielona na trzy etapy. Pierwszym etapem jest identyfikowanie komponentów w ramach, którego pierwszą wykonywaną przez analityka systemowego czynnością jest utworzenie modelu typów (rys. 2). Jedyńm artefaktem wejściowym jest model dziedzinowy. Model dziedzinowy służy do zdefiniowania pojęć użytych podczas budowy systemu informatycznego. Definiowane pojęcia pochodzą z konkretnej dziedziny lub dziedzin, dla której projektowany jest system informatyczny. Mimo, iż model dziedzinowy jest przedstawiany przy użyciu diagramów klas, nie jest on modelem oprogramowania, lecz modelem informacyjnym z dziedziny omawianego problemu. Jedyńm artefaktem wyjściowym jest początkowy model typów. Początkowy model typów jest modelem specyfikacji. Tworzony jest przy użyciu diagramów klas, a wiele klas posiada takie same nazwy jak klasy z modelu dziedzinowego. Jednak klasy w modelu typów reprezentują systemową reprezentację wybranego pojęcia, rozpatrując je bardziej dokładnie i bardziej precyzyjnie definiując.



Rys. 2. Diagram aktywności opisujący czynność tworzenia modelu typów
Fig. 2. Activity 'develop type model' described in activity diagram

W ramach czynności „utworzenie modelu typów” wykonywany jest jeden krok „wyznaczenie poprawnego zasięgu modelu typów” (dalej WPZMT). Główne zadanie, jakie postawiono analitykowi w ramach kroku WPZMT, jest utworzenie na bazie modelu dziedziny początkowego modelu typów. Na początku kroku następuje przekopiowanie modelu pojęciowego a następnie dodanie do niego lub usunięcie z niego klas reprezentujących typy, w celu zdefiniowania poprawnego zasięgu modelu. Głównym kryterium, jakie jest brane pod uwagę podczas dodawania lub usuwania klas, jest to czy informacja opisana w modelu pojęciowym przedsiębiorstwa ma być przechowywana przez projektowany system.

Należy przyjąć zgodnie z wytycznymi metodyki RUP, iż krok WPZMT będzie wykonywany wielokrotnie przy każdej kolejnej iteracji, dlatego należy przyjąć kryterium, przy którym krok WPZMT jest zakończony. Stosując język klauzul można opisać kryteria podejmowania decyzji dla kroku WPZMT.

W celu czytelniejszego przedstawienia opisywanego problemu, klauzule 1a, 2a i 3a zostały zapisane w notacji infiksowej. Dodatkowo każda klauzula (1b, 2b, 3b) została zapisana także w notacji prefiksowej – ułatwiając wykorzystanie języka przetwarzania symbolicznego.

x jest początkowym modelem typów $\leftarrow$
 y jest modelem dziedziny,
 y jest przodkiem x ,
 x jest modelem o poprawnym zasięgu (1a)

$PoczątkowyModelTypow(x) \leftarrow$
 $ModelDziedziny(y),$
 $Przodek(y, x),$
 $ModelOPoprawnymZasięgu(x)$ (1b)

Formuła atomowa $Przodek$ oraz klauzule $ModelDziedziny$ i $ModelOPoprawnymZasięgu$, implikują pojedynczą konkluzję $PoczątkowyModelTypowPrzedsięwzięcia$. Klauzula $PoczątkowyModelTypowPrzedsięwzięcia$ definiuje regułę wyznaczenia artefaktu wyjściowego dla kroku WPZMT.

y jest modelem dziedziny $\leftarrow$
 y definiuje pojęcia z dziedziny implementowanego problemu,
 y definiuje związki,
 y jest diagramem klas (2a)

$ModelDziedziny(y) \leftarrow$
 $DefiniujePojeciaZDziedzinyImplementowanegoProblemu(y),$
 $DefiniujeZwiązki(y),$
 $DiagramKlas(y)$ (2b)

Trzy formuły atomowe $DefiniujePojeciaZDziedzinyImplementowanegoProblemu$, $DefiniujeZwiązki$, $DiagramKlas$ implikują klauzulę $ModelDziedziny$, która jest używana w procesie wnioskowania w klauzuli 1b.

x jest modelem o poprawnym zasięgu $\leftarrow$
 x gdy definiuje jedynie te pojęcia z dziedziny implementowanego problemu, które zostaną zaimplementowane przez system,
 x definiuje wszystkie związki zachodzące pomiędzy bytami (3a)

$ModelOPoprawnymZasięgu(x) \leftarrow$
 $DefiniujeWszystkiePojeciaImplementowanePrzezSystem(x),$
 $DefiniujeWszystkieZwiązkiPomiedzyBytami(x)$ (3b)

Formuły atomowe $DefiniujeWszystkiePojeciaImplementowanePrzezSystem$, $DefiniujeWszystkieZwiązkiPomiedzyBytami$ implikują klauzulę $ModelOPoprawnymZasięgu$, która również jest używana w klauzuli 1b.

Krok WPZMT, można przedstawić formie algorytmu:

```

INPUT: model dziedziny (mdXMI)
OUTPUT: początkowy model typów (pmtXMI)

IF mdXMI IS modelem dziedziny
THEN
  1) przekopiuj mdXMI i oznacz jako pmtXMI
  2) uzupełnij klasy w pmtXMI o atrybuty i typy
  3) uściśl i uzupełnij powiązania w pmtXMI
  4) zdefiniuj w pmtXMI poprawny zasięg
  5) wykonaj klauzulę PoczątkowyModelTypow(pmtXMI)
END IF

```

W celu zwiększenia jednoznaczności przedstawionego algorytmu i zastosowanych oznaczeń wprowadzono następujące definicje:

Definicja 1. Model dziedziny (mdXMI) jest dokumentem XML [8] przechowującym przy użyciu znaczników informacje zdefiniowane jako dwójka: $mdXMI = (klasy, powiązania)$, gdzie: klasy są skończonym zbiorem elementów $klasa = \{nazwa\}$, powiązania są skończonym zbiorem elementów $powiązanie = \{rola, liczebność\}$.

Definicja 2. Początkowy model typów (pmtXMI) jest dokumentem XML, przechowującym przy użyciu znaczników informacje zdefiniowane jako dwójka: $pmtXMI = (klasy, powiązania)$, gdzie: klasy są skończonym zbiorem elementów $klasa = \{nazwa, atrybuty, atrybuty sparometrizedowane\}$, powiązania są skończonym zbiorem elementów $powiązanie = \{nazwa, rola, liczebność, agregacja\}$.

3. Podsumowanie

W prezentowanym artykule sformalizowano przykładową czynność dokonywaną w metodyce CD przy użyciu języka klauzul. Dzięki zastosowaniu notacji infiksowej języka klauzul, realizowane podczas poszczególnych czynności wnioskowanie jest czytelne dla człowieka. Natomiast dzięki zastosowaniu notacji prefiksowej, można wykorzystać opisane reguły podczas budowy systemu wspomagającego decyzje analityka systemowego.

4. Literatura

- [1] Chessman J., Daniels J.: Komponenty w UML, Wydawnictwa Naukowo-Techniczne 2004.
- [2] D'Souza D. F., Wills A. C.: Objects, Components, and Frameworks with UML: The Catalysis Approach, Addison-Wesley 1998.
- [3] Gratkowski T.: Cykl życia projektu informatycznego, Pierwsza Konferencja Naukowa 'Informatyka - sztuka czy rzemiosło', Oficyna Wydawnicza Uniwersytetu Zielonogórskiego 2004, 113–118.
- [4] Kowalski R.: Logika w rozwiązywaniu zadań, Wydawnictwa Naukowo-Techniczne 1989.
- [5] Kruchten P.: The Rational Unified Process An Introduction, Computer software-Development, Addison-Wesley, 2000.
- [6] OMG, Software Process Engineering Metamodel (SPEM), v1.1, Object Management Group, January 2005.
- [7] OMG, Unified Modeling Language: Superstructure, v2.0, Object Management Group, July 2005.
- [8] OMG, XML Metadata Interchange (XMI), v2.1, Object Management Group, September 2005.